UCS 1617 – MINI PROJECT
STUDENTS INFORMATION SYSTEM

PACKAGE & COMPONENT DIAGRAM

TEAM MEMBERS :

- SHRINISHA N (18 5001 148)
- SRIPRABHA AR (18 5001 167)
- SUBA SHREE V S (18 5001 171)

UCS 1617 – MINI PROJECT
STUDENTS INFORMATION SYSTEM

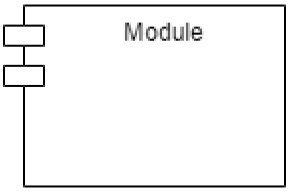
9.1Aim :

To construct a Component diagram & Deployment diagram for an Student Information system using UML notations.

9.2UML Notations for Component Diagram:

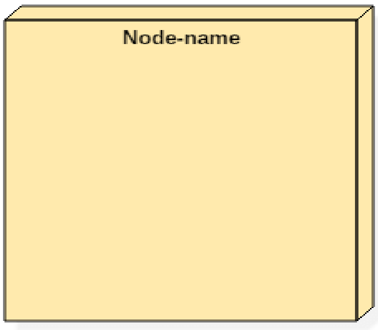
1.Component:

An entity required to execute a stereotype function. A component provides and consumes behaviour through interfaces, as well as through other components. Think of components as a type of class



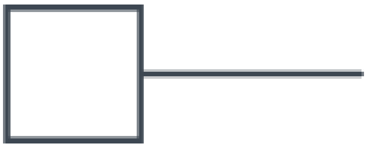
2.Node:

Represents hardware or software objects, which are of a higher level than components.



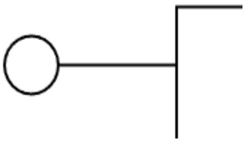
3.Port:

Specifies a separate interaction point between the component and the environment. Ports are symbolized with a small square.



4.Provided Interface:

A straight line from the component box with an attached circle. These symbols represent the interfaces where a component produces information used by the required interface of another component.



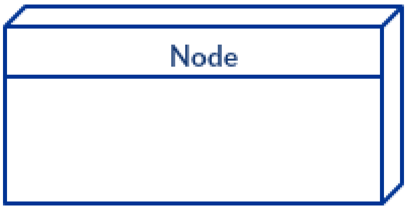
5.Required Interface:

A straight line from the component box with an attached half circle (also represented as a dashed arrow with an open arrow). These symbols represent the interfaces where a component requires information in order to perform its proper function

9.3UML for Deployment Diagram:

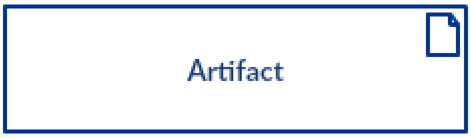
1.Nodes:

A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability. A set of components may reside on a node and may also migrate from node to node. Graphically, a node is rendered as a cube, usually including only its name.



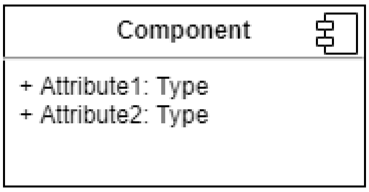
2.Artifacts:

Artifacts are concrete elements that are caused by a development process. Examples of artifacts are libraries, archives, configuration files, executable files etc.



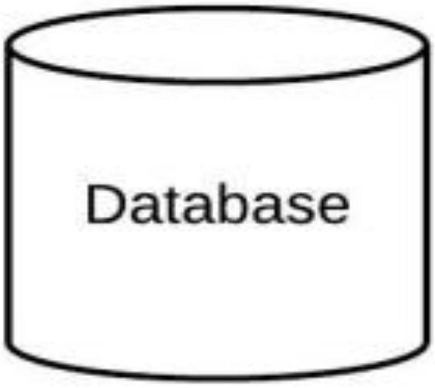
3.Component:

An entity required to execute a stereotype function. A component provides and consumes behaviour through interfaces, as well as through other components. Think of components as a type of class



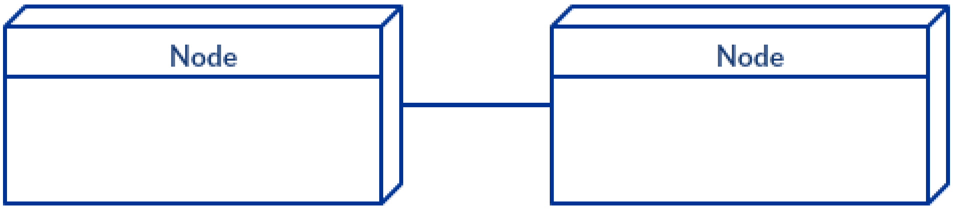
4.Database:

Databases represent any data stored by the deployed system. In some instances, you'll see a database represented as just another node, but sometimes you will see this shape as a database.

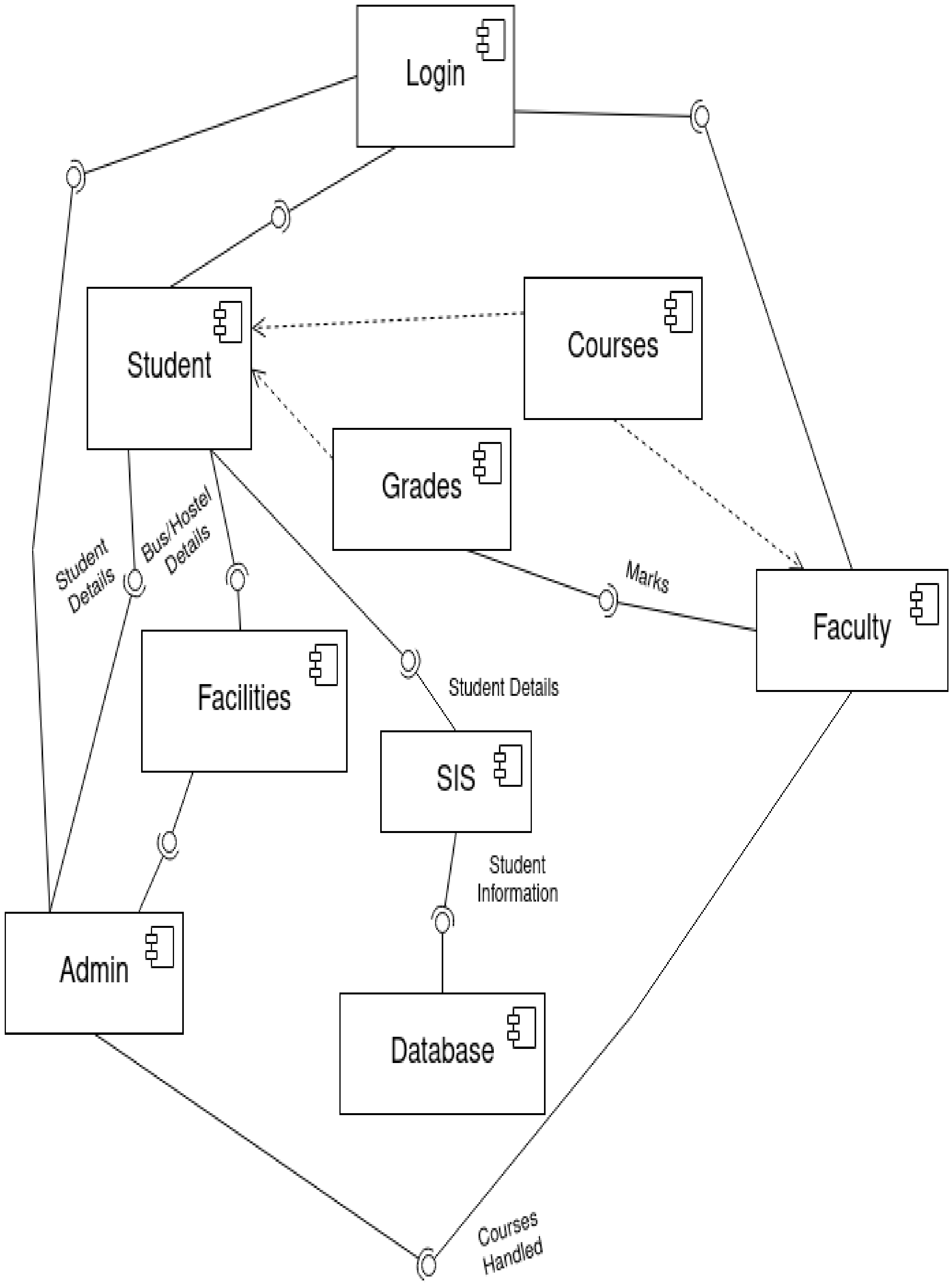


5.Communication Path:

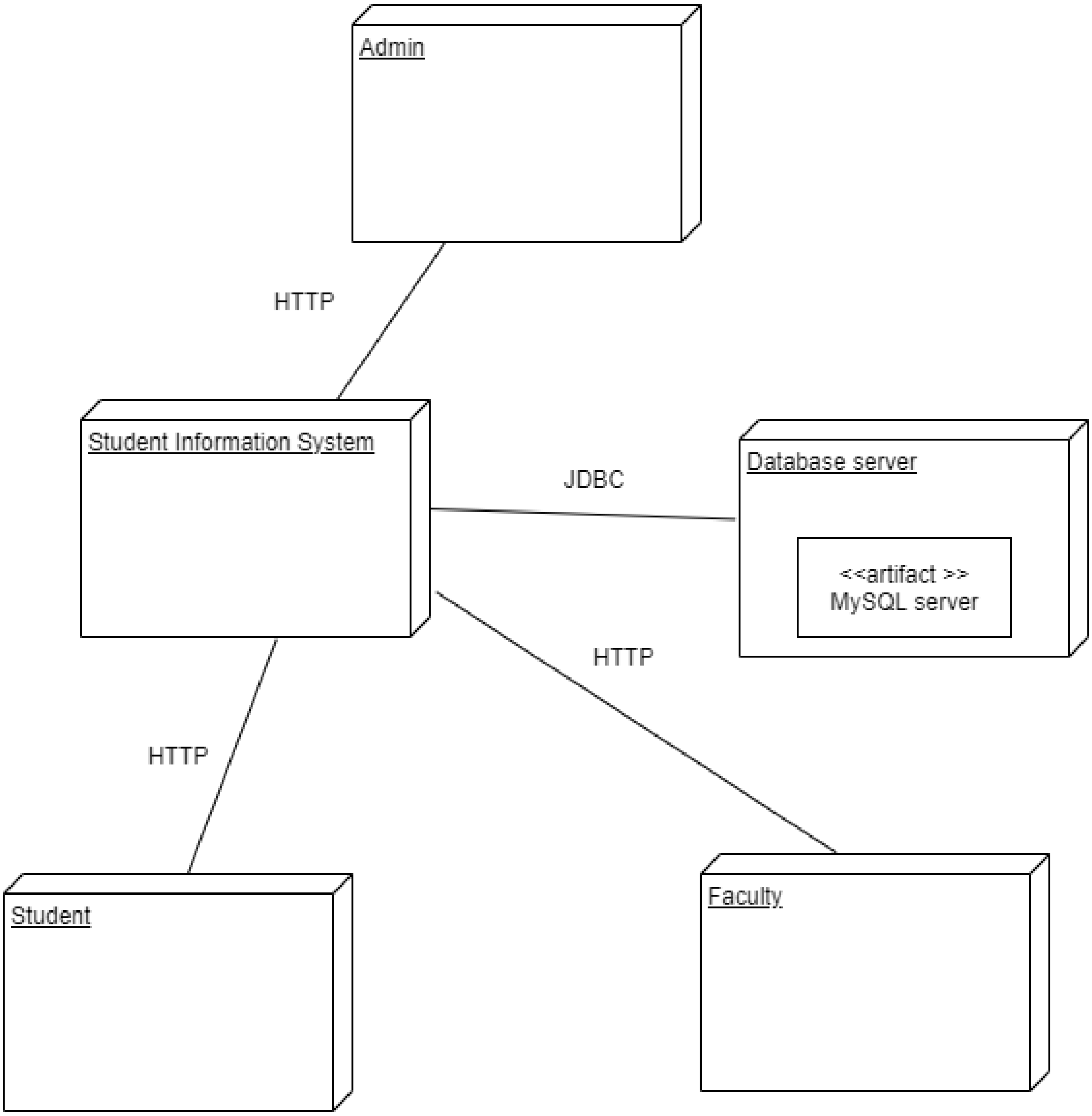
This is represented by a solid line between two nodes. It shows the path of communication between nodes.



9.4 COMPONENT DIAGRAM:-



9.5 DEPLOYMENT DIAGRAM:-


Admin
HTTP
Student Information System
JDBC
Database server
<<artifact >>
MySQL server
HTTP
HTTP
Student
Faculty

Documentation:

A component diagram in UML gives a bird’s-eye view of your software system. Component diagrams can describe software systems that are implemented in any programming language or style. The purpose of a component diagram is to show the relationship between different components in a system. Component refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system. A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system